

Finding Consensus in Multi-Agent Networks Using Heat Kernel Pagerank

Olivia Simpson¹ and Fan Chung²

Abstract—We present a new and efficient algorithm for determining a consensus value for a network of agents. Different from existing algorithms, our algorithm evaluates the consensus value for very large networks using heat kernel pagerank. We consider two frameworks for the consensus problem, a weighted average consensus among all agents, and consensus in a leader-following formation. Using a heat kernel pagerank approximation, we give consensus algorithms that run in time sublinear in the size of the network, and provide quantitative analysis of the tradeoff between performance guarantees and error estimates.

I. INTRODUCTION

The problem of consensus among multi-agent systems has wide applications in situations where members of a distributed network must agree. For example, the communication, feedback, and decision-making between distinct unmanned aerial vehicles (UAVs) [18], [10] is closely related to the consensus problem. In addition to UAVs, the distributed coordination of networks has important implications in cooperative control of distributed sensor networks [9], flocking and swarming behavior [27], and communication congestion control [21]. Further, they form the foundation of the field of distributed computing [14]. The consensus problem is studied in [19], and several variations and extensions are examined by [20], [17], [15], [4].

We consider the classical model (see [19]) of agents with fixed, bidirectional communication channels and associated state. State changes occur continuously, influenced by communication with neighbors. A consensus algorithm is a continuous time protocol that specifies the information exchange between agents and provides a mechanism for systematically computing the consensus value, a unanimous state and an equilibrium of the system. In this paper, we focus on an efficient method for approximating the state values in a network in which agents reach consensus by following a linear protocol. We give algorithms for two different frameworks. The first computes a global consensus value involving all the agents in the network and runs in time sublinear in the size of the network. The second is a local algorithm to compute a consensus

value for a subset of agents under external influence, and runs in time sublinear in the size of the specified subset. In both, the consensus value returned is within an error bound of $O(\epsilon)$ for a given $0 < \epsilon < 1$.

Our algorithm involves solving a linear system by approximating the heat kernel pagerank of the network and relies on spectral analysis. The tools we present are relevant to numerous graph problems, including partitioning and clustering algorithms [3], [25], [16], flow and diffusion modeling [24], electrical network theory [5], and regression on graphs [2].

A. Previous Work

a) The consensus problem: In [19], Olfati-Saber and Murray design a linear protocol for agents to reach a consensus value which is an average of initial states. They consider a network of agents as an undirected graph and use the Laplacian potential, defined in terms of the graph Laplacian (to be defined in Section II-B), as a measure of disagreement among nodes. With this tool, they transform the the problem of reaching consensus to that of minimizing the Laplacian potential.

An alternate formulation given in [10] abides by a linear protocol which favors the values of more highly connected nodes. In this way, agents which are more visible will have more of an impact on the group decision. Yet another variation is consensus in a leader-following formation, in which a set of agents called leaders abide by individual protocol but continue to influence to rest of the network. This problem has been studied in [23], [17].

b) Laplacian linear systems: Fast methods for solving systems of linear equations gained awareness with the nearly-linear time solver of [26]. Their algorithm implements a recursive procedure for sparsifying a graph related to the coefficient matrix so that solving the system is easy at the base of the recursion. This work was improved in [11], [12] with a higher quality sparsifier which reduced the depth of recursion. A parallel solver for SDD systems is given in [22] which runs in polylogarithmic time and nearly-linear work, an improvement to previous bounds.

The methods in this paper are closely related to previous work on approximating the discrete Green's function (or pseudo-inverse of the Laplacian matrix) [8]. They give an algorithm for solving Laplacian linear systems with a boundary condition on a subset

¹Dept. of Computer Science, University of California, San Diego, La Jolla, CA osimpson@ucsd.edu

²Dept. of Mathematics, University of California, San Diego, La Jolla, CA fan@ucsd.edu

of vertices that improves previous time bounds by using the method of heat kernel.

B. Main Results

In the model we consider, the communication protocol followed by the agents in the network forms a linear system of equations, and the solution to the linear system is the state of the network as a function of time. Thus, computing the consensus value involves solving a linear system.

We consider two forms of consensus. In the group consensus framework, we seek a consensus value that is a weighted average of initial states of the system, with weights proportional to node degrees. In this case, our algorithm computes the consensus value by approximating the state vector corresponding to the equilibrium of the system in sublinear time. In the local framework, a subset of agents imposes an external influence on an adjacent subset. The consensus achieved in this case is referred to as a leader-following consensus. Our algorithm for computing leader-following consensus on the subset of followers involves sampling vectors that approximate the equilibrium state vector, and runs in sublinear time.

Specifically, our contributions are:

- 1) We give a new algorithm for approximating the state of a system in a weighted average consensus framework to within a multiplicative factor $(1 + \varepsilon)$ and additive term $O(\varepsilon)$ in time $O\left(\frac{\log(\varepsilon^{-1}) \log n}{\varepsilon^3 \log \log(\varepsilon^{-1})}\right)$ where n is the size of the network. We call this algorithm AVGCONSENSUS and present it in Section III-B.
- 2) We give a new algorithm for approximating the state of a subset of agents in a leader-following consensus framework to within a multiplicative factor $(1 + \varepsilon)$ and additive term $O(\varepsilon)$ in time $O\left(\frac{(\log s)^2 (\log(\varepsilon^{-1}))^2}{\varepsilon^5 \log \log(\varepsilon^{-1})}\right)$, where s is the size of the subset of followers. We call this algorithm LF-CONSENSUS and present it in Section IV.

Our sublinear time algorithms for computing consensus value rely on the efficiency of an algorithm for approximating heat kernel pagerank. Heat kernel pagerank is introduced in detail in [6] and [7] as a variant of Personalized PageRank [1]. The heat kernel pagerank approximator is introduced Section III-B and in more detail in [8].

II. PRELIMINARIES

A. Networked Multi-Agent Systems

A dynamic multi-agent system is given by a tuple $G_x = (G, x)$ where x is the state of the system and G is the communication network topology, represented by a graph. Namely, each agent is represented by a node and the communication network between agents is represented by the edge set $E(G)$. Let $x_i \in \mathbb{R}$ be

a real scalar value assigned to v_i such that $x(t) = (x_1(t), \dots, x_n(t))^T$ denotes the state at time t .

For an undirected graph $G = (V, E)$ of size $|V| = n$, let the nodes of V be arbitrarily indexed by index set $\mathcal{I} = \{1, \dots, n\}$ such that $E \subset \mathcal{I} \times \mathcal{I}$. For a node $v_i \in V$, let $N_i = \{v_j \in V | (i, j) \in E\}$ be the set of *neighbors* of v_i and let $d_i = |N_i|$ be the *degree* of v_i . Two nodes v_i, v_j are said to *agree* if and only if $x_i = x_j$. The goal of consensus is to minimize the total disagreement among nodes.

Definition II.1 (Consensus). Let the value of nodes x be the solution to the equation

$$\dot{x} = f(x, u), \quad x(0) \in \mathbb{R}^n. \quad (1)$$

Let $\chi: \mathbb{R}^n \rightarrow \mathbb{R}$ be an operator on $x = (x_1, \dots, x_n)^T$ that generates a decision value $\chi(x)$. Then we say all nodes of the graph have reached *consensus with respect to χ* in finite time $T > 0$ if and only if all nodes agree and $x_i(T) = \chi(x(0)) \forall i \in \mathcal{I}$. We call $\chi(x) := \chi(x(0))$ the *consensus value*.

One notion of consensus is a *weighted average consensus*, given by

$$\chi_w(x) = \frac{\sum_i d_i x_i}{\sum_i d_i}.$$

We show (Theorem III.1) that any connected undirected graph globally asymptotically reaches weighted average consensus when each node applies the distributed linear protocol

$$u_i(t) = 1/d_i \sum_{j \in N_i} (x_j(t) - x_i(t)). \quad (2)$$

We assume G is connected for the remainder of the paper.

B. Graph Laplacians and Heat Kernel

In this work, we consider graphs which are weight-normalized so that every entry of the weighted adjacency matrix A is $a_{ij} \in \{0, 1\}$, and the unordered pair $(i, j) \in E(G)$ if and only if $a_{ij} = 1$. Let D denote the diagonal degree matrix $D(i, i) = d_i$.

The *Laplacian* is defined $\mathcal{L} = D^{-1/2}(D - A)D^{-1/2}$. Let Δ be the graph matrix $\Delta = I - D^{-1}A$. We call Δ the *Laplace operator*. We note that Δ is similar to the matrix \mathcal{L} .¹

The *heat kernel* of a graph is a solution to the heat differential equation

$$\frac{\partial u}{\partial t} = -\Delta u.$$

The heat kernel can be formulated in the context of random walks on graphs. Consider the transition

¹The Laplacian used in [19] is the matrix $L = D - A$, a common variation.

probability matrix associated to a random walk given by $P = D^{-1}A$. Then heat kernel is defined:

$$H_t = e^{-t\Delta} = \sum_{k=0}^{\infty} \frac{(-t)^k}{k!} \Delta^k \quad (3)$$

$$= e^{-t(I-P)} = \sum_{k=0}^{\infty} e^{-t} \frac{t^k}{k!} P^k. \quad (4)$$

The following similarity of the heat kernel, \mathcal{H}_t , is of interest for its symmetry. Using definition (3),

$$\begin{aligned} \mathcal{H}_t &= \sum_{k=0}^{\infty} \frac{t^k}{k!} D^{1/2} \Delta^k D^{-1/2} \\ &= \sum_{k=0}^{\infty} \frac{t^k}{k!} D^{1/2} (D^{-1/2} \mathcal{L} D^{1/2})^k D^{-1/2} \\ &= \sum_{k=0}^{\infty} \frac{t^k}{k!} \mathcal{L}^k = e^{-t\mathcal{L}}. \end{aligned}$$

Heat kernel pagerank is a row vector determined by two parameters; $t \in \mathbb{R}^+$, and a preference row vector $f \in \mathbb{R}^n$. It is given by the following equation:

$$\rho_{t,f} = f H_t = \sum_{k=0}^{\infty} e^{-t} \frac{t^k}{k!} f P^k.$$

Specifically, it is an exponential sum of random walks generated from a starting vector, $f \in \mathbb{R}^n$. As an added benefit, heat kernel pagerank simultaneously satisfies the heat equation with the rate of diffusion controlled by the parameter t . Both properties are powerful tools in consensus problems.

III. HEAT KERNEL PAGERANK FOR WEIGHTED-AVERAGE CONSENSUS

In this section we present a linear consensus protocol for a dynamic network and show how to compute a weighted average consensus for the protocol using heat kernel pagerank.

We first recall some principles of control theory. Consider the system with controls as in (1). A point x_e is an *equilibrium point* of the system if $f(x_e, u) = 0$, and x_e is an equilibrium point if and only if $x(t) = x_e$ is a trajectory. The system is *globally asymptotically stable* if, for every trajectory $x(t)$, $x(t) \rightarrow x_e$ as $t \rightarrow \infty$. To check this, two conditions are sufficient.

Definition III.1 (Global asymptotic stability). A system is *globally asymptotically stable* if

- 1) it is stable in the Lyapunov sense, and
- 2) the equilibrium x_e is convergent, i.e., for every $\varepsilon > 0$, there is some time T such that

$$\|x(0) - x_e\| < \delta \quad \text{means} \quad \|x(t) - x_e\| < \varepsilon$$

for every time $t > T$.

In particular, when considering a time-invariant linear state space model $\dot{x} = -Mx$, for some matrix M , condition 1 is satisfied if M is positive semidefinite.

A. Consensus and the Laplacian

Consider the network of integrator agents with dynamics $\dot{x}_i = u_i$ where each agent applies the distributed linear protocol (2). We can characterize the dynamics of the system by the Laplace operator for the underlying graph, as described by the following theorem:

Theorem III.1. *Let G_x be a dynamic multi-agent system and suppose each node of G applies the distributed linear protocol (2). Then the value of x at time t is given by the solution to the system*

$$\dot{x}(t) = -\Delta x(t), \quad x(0) \in \mathbb{R}^n. \quad (5)$$

Additionally, this protocol globally asymptotically reaches a weighted average consensus.

Proof. Let x_e be an equilibrium of the system $\dot{x} = -\Delta x$. Then by definition of equilibrium, $\Delta x_e = 0$ and therefore x_e is a right eigenvector associated to the eigenvalue $\lambda = 0$. In particular x_e is in the null space of Δ . Since G is connected, Δ has exactly one zero eigenvalue. Upon consideration, we see that the corresponding eigenvector is $\mathbf{1}$, the all-one's vector, as the row sums of Δ are all exactly zero. Thus, $x_e = \alpha \mathbf{1}$ for some $\alpha \in \mathbb{R}$. Now, note that $\sum_i u_i = 0$ for the protocol (2), and so the weighted average value $\chi_w(x(t))$, determined by $u(t)$, is in fact invariant with respect to t . In other words, $\chi_w(x(0)) = \chi_w(x_e)$, and

$$\chi_w(x_e) = \frac{\sum_i d_i (x_e)_i}{\sum_i d_i} = \alpha.$$

Therefore this equilibrium is in fact the weighted average of the initial values of the nodes, and all nodes reach this value. Also, as the system is time-invariant, the system is stable since Δ is positive semidefinite.

By Definition III.1, the Theorem is proved. \square

Now we can summarize the state of the system with a single heat kernel pagerank vector.

Theorem III.2. *Let G_x be a dynamic multi-agent system and suppose each node of G applies the distributed linear protocol (2). Let D be the diagonal degree matrix of G . Then the state of the system is given by*

$$x(t) = \rho_{t,f} D^{-1}, \quad f = x(0)^{tr} D, \quad (6)$$

where M^{tr} denotes the transpose.

Proof. The solution to (5) is the evolving state of the system. This solution is

$$x(t) = e^{-t\Delta} x(0) = H_t x(0). \quad (7)$$

Using the symmetrized version of heat kernel,

$$\begin{aligned} x(t) &= (D^{-1/2} \mathcal{H}_t D^{1/2}) x(0) \\ x(t)^{tr} &= x(0)^{tr} D^{1/2} (D^{1/2} H_t D^{-1/2}) D^{-1/2} \\ x(t)^{tr} &= (x(0)^{tr} D) H_t D^{-1/2}, \end{aligned} \quad (8)$$

where line 8 uses the symmetry of D and \mathcal{H} . Thus, the values $x(t)$ given by (7) are related to the heat kernel pagerank vector $\rho_{t,f}$ with preference vector $f = x(0)^T D$. \square

To compute the equilibrium state at which all agents reach consensus, we know that time $T = O(1/\lambda_1)$ is an upper bound. Figure 1b depicts the results of computing weighted average consensus with heat kernel pagerank as in Theorem III.2 with different values for t . The network is an undirected social network of dolphins [13] with initial state values randomly chosen from the interval $(0, 1)$ (Figure 1a). The chart plots total disagreement $\|\delta\|$ for disagreement vector $\delta(t) = x(t) - \chi_w(x)\mathbf{1}$, where $x(t) = \rho_{t,f} D^{-1}$ for $f = x(0)^T D$. The vertical line corresponds to $t = 1/\lambda_1$.

B. An Algorithm for Computing Consensus Value Using Approximate Heat Kernel Pagerank

Our weighted average consensus algorithm uses an algorithm for approximating heat kernel pagerank as a subroutine. We use the following definition of an approximate heat kernel pagerank.

Definition III.2. Let $f \in \mathbb{R}^n$ be a vector over nodes of a graph $G = (V, E)$ and let $\rho_{t,f}$ be the heat kernel pagerank vector over G according to t and f . Then we say that $v \in \mathbb{R}^n$ is an ε -approximate heat kernel pagerank vector if

- 1) for every node $v_i \in V$ in the support of v , $(1 - \varepsilon)\rho_{t,f}[i] - \varepsilon \leq v[i] \leq (1 + \varepsilon)\rho_{t,f}[i]$, and
- 2) for every node with $v[i] = 0$, it must be that $\rho_{t,f}[i] \leq \varepsilon$.

Theorem III.3 (Weighted Average Consensus in Sublinear Time). *Let G_x be a dynamic n -agent system and suppose each node of G applies the distributed linear protocol (2). Then the state of the system can be approximated to within a multiplicative factor of $(1 + \varepsilon)$ and an additive term of $O(\varepsilon)$ for any $0 < \varepsilon < 1$ in time $O(\frac{\log(\varepsilon^{-1}) \log n}{\varepsilon^3 \log \log(\varepsilon^{-1})})$.*

We call the algorithm AVGCONSENSUS. The algorithm makes a call to APPROXHKPR, an extension of the algorithm presented in [8] for quickly computing an approximation of a restricted heat kernel pagerank vector. For the sake of completeness, the algorithm and a summary of the results of [8] are given at the end of this section.

Proof of Theorem III.3. First, the ε -approximate vector x returned by APPROXHKPR is an approximation of the true state by Theorem III.2. Thus we have left to verify the approximation guarantee and the running time. The total running time is dominated by the heat kernel pagerank approximation, which

is $O(\frac{\log(\varepsilon^{-1}) \log n}{\varepsilon^3 \log \log(\varepsilon^{-1})})$ by Theorem III.4, below. Theorem III.4 also verifies the approximation guarantee. \square

Algorithm III.1: AVGCONSENSUS(G, x, t, ε)

comment: input:

comment: G as the $(0, 1)$ -adjacency matrix

comment: x , initial state vector

comment: $t \in \mathbb{R}^+$

comment: $0 < \varepsilon < 1$, error parameter

comment: output: $x(t)$

$D \leftarrow$ diagonal matrix of rowsums(G)

$f \leftarrow x^T D$

$y \leftarrow \text{APPROXHKPR}(G, t, f, \varepsilon)$

return (yD^{-1})

C. A Sublinear Time Heat Kernel Pagerank Approximation Algorithm

The analysis for approximating heat kernel pagerank follows easily from that for a restricted heat kernel pagerank vector by considering the entire vertex set rather than a subset. We refer the reader to [8] for a more complete description.

Algorithm 1 APPROXHKPR(G, t, f, ε)

input: a graph G , $t \in \mathbb{R}^+$, preference vector $f \in \mathbb{R}^n$, error parameter $0 < \varepsilon < 1$.

output: ρ , an ε -approximation of $\rho_{t,f}$.

initialize 0-vector ρ of dimension n , where $n = |V|$

$r \leftarrow \frac{16}{\varepsilon^3} \log n$

$K \leftarrow \frac{\log(\varepsilon^{-1})}{\log \log(\varepsilon^{-1})}$

$f' \leftarrow f / \|f\|_1$ \triangleright normalize f to be a probability distribution vector

for r iterations **do**

choose a starting vertex u according to the distribution vector f'

Start

simulate a P random walk where k steps are taken with probability $e^{-t} \frac{t^k}{k!}$ and $k \leq K$

let v be the last vertex visited in the walk

$\rho[v] \leftarrow \rho[v] + 1$

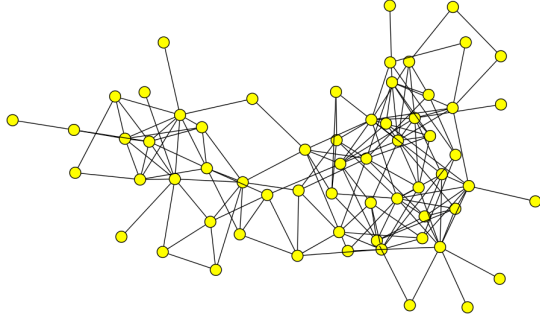
End

end for

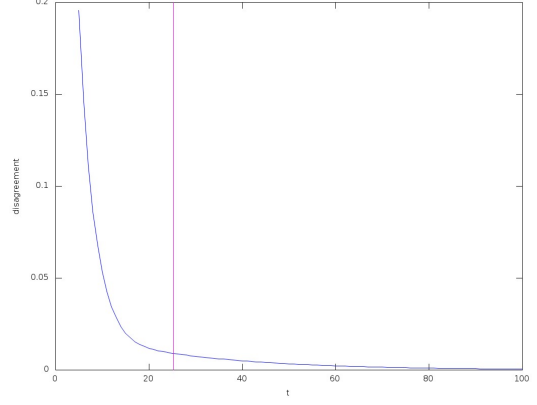
$\rho \leftarrow 1/r \cdot \rho$

return $\|f\|_1 \cdot \rho$

Theorem III.4. *Let G be a graph, $t \in \mathbb{R}^+$, and $f \in \mathbb{R}^n$. Then, the algorithm APPROXHKPR(G, t, f, ε) outputs*



(a) Dolphin social network.



(b) Total disagreement over varying times t . Disagreement is computed in terms of the weighted average consensus $\chi_w(x_0)$. The red line denotes the data point for $t = 1/\lambda_1$.

Fig. 1: Weighted average consensus convergence results.

an ε -approximate vector $\hat{p}_{t,f}$ of the heat kernel pagerank $p_{t,f}$ for $0 < \varepsilon < 1$ with probability at least $1 - \varepsilon$. The running time of ApproxHKPR is $O\left(\frac{\log(\varepsilon^{-1}) \log n}{\varepsilon^3 \log \log(\varepsilon^{-1})}\right)$.

IV. HEAT KERNEL PAGERANK FOR CONSENSUS IN LEADER-FOLLOWING FORMATIONS

In this section we consider a multi-agent network in which a certain subset of agents $l \subset V$ are *leaders*, and the rest $f = V \setminus l$ are dubbed *followers*. In this scenario, leaders will adjust their values according to individual protocol, while followers in the system adjust according to communication channels as usual. The consensus goal in this case is a *leader-following consensus*, in which all agents agree on a value by following the leaders.

Let u^f denote the protocol among the set of followers and let u^l denote the control dictated by the leaders and influencing the followers. Similarly, let x^f denote the state of the followers and x^l denote the state of the leaders. The vectors x^f and x^l can be understood as the usual state vector x restricted to following and leading agents, respectively. Then we have the following definition.

Definition IV.1 (Leader-following consensus). A *leader-following consensus* of a system is achieved if for every agent v_i there is a local protocol u_i such that $x_i(T) = \chi_{lf}(x(0))$ for some finite time $T > 0$ and some operator $\chi_{lf} : \mathbb{R}^n \rightarrow \mathbb{R}$. In this case, we call the value $\chi_{lf}(x(0))$ the leader-following consensus value.

For the protocol

$$u_i(t) = 1/d_i \sum_{j \in N_i} \left(\sqrt{\frac{d_i}{d_j}} x_j(t) - x_i(t) \right), \quad (9)$$

the value for x is given by the dynamics

$$\dot{u}(t) = \dot{x}(t) = -\mathcal{L}x(t).$$

We let the followers abide by protocol (9).

Let \mathcal{L}_f be the Laplacian \mathcal{L} restricted to rows and columns corresponding to the followers, and \mathcal{L}_{fl} be \mathcal{L} with rows restricted to the followers and columns restricted to the leaders. Then the dynamics of the followers can be summarized by:

$$\dot{x}^f(t) = -\mathcal{L}_f x^f(t) - \mathcal{L}_{fl} u^l(t).$$

Since \dot{x}^f is control of the subnetwork induced by the group of followers, this can be rewritten

$$\begin{aligned} u^f &= -\mathcal{L}_f x^f - \mathcal{L}_{fl} u^l \quad \text{or alternatively,} \\ x^f &= -\mathcal{L}_f^{-1} u^f + \mathcal{L}_f^{-1} \mathcal{L}_{fl} u^l. \end{aligned}$$

Indeed, as long as the subgraph induced by the subset of followers is connected, the inverse \mathcal{L}_f^{-1} exists. We have arrived at the following.

Theorem IV.1. Let G_x be a dynamic multi-agent system with proper subsets of leaders, $l \subset V$, and followers, $f = V \setminus l$, such that the induced subgraph on f is connected. Suppose the followers apply the protocol (9), and suppose the leaders apply some individual protocol $u_i = f(x_i)$ dictated only by that leader's state. Then the followers' state values x^f at time t are given by the solution to the system

$$\mathcal{L}_f x^f(t) = b(t), \quad \text{where} \quad b(t) = -(u^f(t) + \mathcal{L}_{fl} u^l(t)).$$

An efficient algorithm called GreensSolver for solving linear systems $\mathcal{L}_f x^f = b$ with a linear protocol applied to a subset specified by b is given in [8]. They

show that the solution x_f can be computed with the symmetric heat kernel using the relationship

$$\mathcal{L}_f^{-1}b = \int_0^\infty (\mathcal{H}_t)_f dt b, \quad (10)$$

where $(\mathcal{H}_t)_f$ is \mathcal{H}_t with rows and columns restricted to the set f .

The solution $\mathcal{L}_f^{-1}b$ can be approximated by sampling sufficiently many values of $(\mathcal{H}_t)_f b(t)$. Further, it is given that the solution can be approximated in $O\left(\frac{(\log s)^2 (\log(\epsilon^{-1}))^2}{\epsilon^5 \log \log(\epsilon^{-1})}\right)$ time, where s is the size of the subset of followers.

Algorithm IV.1: LFCONSENSUS($G, x, t, f, l, u^l, \epsilon$)

comment: input:

comment: G as the $(0, 1)$

comment: x , initial state vector

comment: $t \in \mathbb{R}^+$

comment: f , subset of followers

comment: l , subset of leaders

comment: u^l , protocol applied by the leaders

comment: $0 < \epsilon < 1$, error parameter

comment: output: $x(t)$

procedure FOLLOWERPROT(G, x, t)

for each $i \in l$

do $u^f[i] \leftarrow u_i(t) = 1/d_i \sum_{j \in N_i} \left(\sqrt{\frac{d_i}{d_j}} x_j(t) - x_i(t) \right)$

return (u^f)

procedure B(t)

$u^f \leftarrow \text{FOLLOWERPROT}(G, x, t)$

$b \leftarrow -(u^f(t) + \mathcal{L}_f l u^l)$

$b \leftarrow B(t)$

$s \leftarrow |f|$

$T \leftarrow s^3 \log(1/\epsilon)$

$N \leftarrow T/\epsilon$

$r \leftarrow \frac{\log s + \log(\epsilon^{-1})}{\epsilon^2}$

initialize a 0 -vector x^f of dimension s

for $i = 0$ **to** r

do $\begin{cases} \text{draw } j \text{ from } [1, N] \text{ uniformly at random} \\ x_i \leftarrow \text{ApproxHK}(G, jT/N, b, f, \epsilon) \\ x^f \leftarrow x^f + x_i \end{cases}$

return $(\frac{1}{r} x D_S^{-1/2})$

The running time and approximation guarantees of LFCONSENSUS follow from the running time of GreensSolver [8], and we have the following:

Theorem IV.2 (Leader-Following Consensus in Sublinear Time). *Let G_x be a dynamic multi-agent system*

with proper subsets of leaders, $l \subset V$, and followers, $f = V \setminus l$, such that the induced subgraph on f is connected. Suppose the followers apply the protocol (9), and suppose the leaders apply some individual protocol $u_i = f(x_i)$ dictated only by that leader's state. Then the state of the system can be approximated to within a multiplicative factor of $(1 + \epsilon)$ and an additive term of $O(\epsilon)$ for any $0 < \epsilon < 1$ in time $O\left(\frac{(\log s)^2 (\log(\epsilon^{-1}))^2}{\epsilon^5 \log \log(\epsilon^{-1})}\right)$, where s is the size of the subset of followers.

V. DISCUSSION

The significance of sublinear running times is scalability. The robustness and efficiency of the algorithms AVGCONSENSUS and LFCONSENSUS are of great importance for networks too large to fit in memory, and the running time/approximation tradeoff allows for appropriate tuning. This is especially notable for local algorithms, which reduce computation over large networks to a small subset. For instance, while the group of leaders may be small in a leader-following framework, the difference in complexity for computing consensus in a leader-following formation as opposed to full group consensus can be significant. The subset of followers influenced by the leaders may be a small portion of the entire graph, so that $s \ll n$, and we are spared work over the entire graph in the case that we are interested in only a small area. In these cases, the gain in running times are valueable.

REFERENCES

- [1] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, pages 107–117, 1998.
- [2] D. Cai, X. He, and J. Han. Spectral regression for efficient regularized subspace learning. In *IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.
- [3] P. K. Chan, M. D. Schlag, and J. Y. Zien. Spectral k-way ratio-cut partitioning and clustering. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 13(9):1088–1096, 1994.
- [4] L. Cheng, Z.-G. Hou, and M. Tan. Reaching a consensus in networks of high-order integral agents under switching directed topology. *arXiv preprint arXiv:1304.3972*, 2013.
- [5] P. Christiano, J. A. Kelner, A. Madry, D. A. Spielman, and S.-H. Teng. Electrical flows, laplacian systems, and faster approximation of maximum flow in undirected graphs. *CoRR*, abs/1010.2921, 2010.
- [6] F. Chung. The heat kernel as the pagerank of a graph. *Proceedings of the National Academy of Sciences*, 104(50):19735–19740, 2007.
- [7] F. Chung. A local graph partitioning algorithm using heat kernel pagerank. *Internet Mathematics*, 6(3):315–330, 2009.
- [8] F. Chung and O. Simpson. Solving linear systems with boundary conditions using heat kernel pagerank. In *Algorithms and Models for the Web Graph*, pages 203 – 219, 2013.
- [9] J. Corts and F. Bullo. Coordination and geometric optimization via distributed dynamical systems. *SIAM Journal on Control and Optimization*, 44(5):1543–1574, 2005.
- [10] J. Fax and R. Murray. Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, 49(9):1465–1476, 2004.

- [11] I. Koutis, G. L. Miller, and R. Peng. Approaching optimality for solving sdd linear systems. In *IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 235–244. IEEE, 2010.
- [12] I. Koutis, G. L. Miller, and R. Peng. A nearly- $m \log n$ time solver for sdd linear systems. In *IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 590–598. IEEE, 2011.
- [13] D. Lusseau, K. Schneider, O. Boisseau, P. Haase, E. Slooten, and S. Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54:396–405, 2003.
- [14] N. A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1996.
- [15] K. L. Moore, T. Vincent, F. Lashhab, and C. Liu. Dynamic consensus networks with application to the analysis of building thermal processes. In *Proceedings of 18th IFAC World Congress, Milane, Italy*, 2011.
- [16] A. Y. Ng, M. I. Jordan, Y. Weiss, et al. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856, 2002.
- [17] W. Ni and D. Cheng. Leader-following consensus of multi-agent systems under fixed and switching topologies. *Systems & Control Letters*, 59(3–4):209–217, 2010.
- [18] R. Olfati-Saber and R. Murray. Graph rigidity and distributed formation stabilization of multi-vehicle systems. In *Proceedings of the 41st IEEE Conference on Decision and Control*, volume 3, pages 2965–2971, 2002.
- [19] R. Olfati-Saber and R. Murray. Consensus protocols for networks of dynamic agents. In *Proceedings of the American Control Conference*, volume 2, pages 951–956, 2003.
- [20] R. Olfati-Saber and R. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, 2004.
- [21] F. Paganini, J. Doyle, and S. Low. Scalable laws for stable network congestion control. In *Proceedings of the 40th IEEE Conference on Decision and Control*, volume 1, pages 185–190, 2001.
- [22] R. Peng and D. A. Spielman. An efficient parallel solver for sdd linear systems. arXiv:1311.3286, November 2013.
- [23] A. Rahmani, M. Ji, M. Mesbahi, and M. Egerstedt. Controllability of multi-agent systems from a graph-theoretic perspective. *SIAM Journal on Control and Optimization*, 48(1):162–186, 2009.
- [24] A. Raj, A. Kuceyeski, and M. Weiner. A network diffusion model of disease progression in dementia. *Neuron*, 73(6):1204–1215, 2012.
- [25] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
- [26] D. A. Spielman and S.-H. Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of Computing*, pages 81–90. ACM, 2004.
- [27] J. Toner and Y. Tu. Flocks, herds, and schools: A quantitative theory of flocking. *Phys. Rev. E*, 58:4828–4858, Oct 1998.